14th Annual International Conference on Industrial Engineering and Operations Management
Dubai, United Arab Emirates (UAE), February 12-14, 2024
Publisher: IEOM Society International, USA DOI: 10.46254/AN14.20240160
Published: February 12, 2024

Enhancing Learning Analytics in Open-Source Software Mailing Archives using Machine Learning and Process Discovery Techniques

Patrick Mukala
Faculty of Engineering and Computer Science
University of Wollongong in Dubai
Dubai, UAE
patrickmukala@uowdubai.ac.ae

Obaid Ullah
Faculty of Engineering and Computer Science
University of Wollongong in Dubai
Dubai, UAE
obaidullah@uowdubai.ac.ae

Abstract

Existing evidence indicates that Free/Libre Open-Source Software (FLOSS) ecosystems offer extensive learning opportunities. Community members actively participate in various activities, both during their interactions with peers and while utilizing these environments. Given that FLOSS repositories contain valuable data on participant interactions and activities, our study focuses on analyzing knowledge exchange and interactions within emails to track learning activities across different phases of the learning process, with a focus on the first phase (Initiation). In this paper, we leverage Natural Language Processing (NLP) and Machine Learning (ML) techniques within a process mining framework. Specifically, we employ NLP techniques to analyze the contents of emails and messages exchanged in these FLOSS repositories to generate event logs for the purpose of modeling learning patterns. Subsequently, we construct corresponding event logs, which serve as input to Disco, the process mining tool, for learning process discovery in these environments. The output comprises visual workflow nets that we interpret as representations of learning activity traces within FLOSS, capturing their sequential occurrences. To enhance the understanding of these models, we incorporate additional statistical details for contextualization and description. This approach enables a nuanced exploration of learning dynamics within FLOSS environments, emphasizing the role of NLP and ML in uncovering valuable insights on how FLOSS participants acquire and exchange knowledge.

Keywords

FLOSS learning processes, Learning Analytics, Mining software repositories, Process Mining, Semantic Search, Machine Learning, Natural Learning Processing, NLP

1. Introduction

Over the past decades, numerous studies conducted have shown evidence that there are substantial learning opportunities in Free/Libre Open Source Software (FLOSS) environments (Sowe and Stamelos 2008; Cerone et al. Fernandes et al. 2014; Mukala et al. 2014; Jaccheri and Osterlie 2007). Some of these environments or platforms that

can be analyzed for learning events include internet relay chats, email messages and CVS software such as Git and Confluence.

Jaccheri and Osterlie (2007) ascertains a long surging interest from a number of higher learning institutions and schools that have considered including and making mandatory participation in activities from forums or other FLOSS environments as a part of their coursework practices in Software Engineering Courses in order to provide a near-real life projects experience. Concurrently, attempts for endeavors evaluating the effectiveness of this approach are also taking shape and mostly converging towards a positive review.

In our previous study (Mukala et al. 2015) we discussed the importance of the leaning that happens in these FLOSS environments and how learning in a software development environment happens in a set number of steps following a framework organized in phases. These phases include Initiation, Progression and Maturation. Several activities are executed in each of these phases by learning participants (Novice and Expert) for knowledge exchange and acquisition. A Novice participant typically is looking for knowledge and advice in a particular topic or concept while an Expert, on the other side, provides the knowledge.

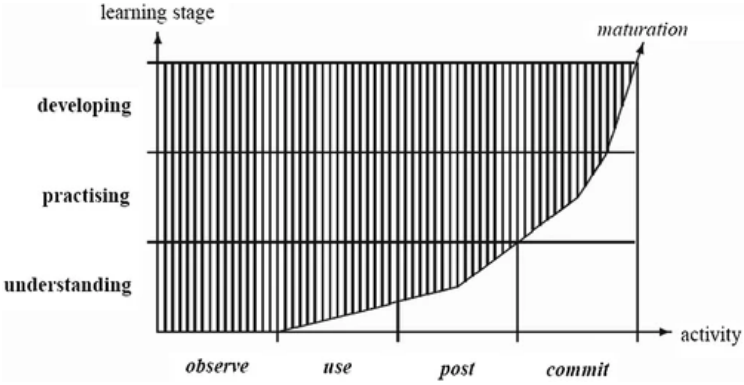

Figure 1. Learning stages and participants' learning progression in OSS communities

In this paper, we extend our contribution from our previous work focused on semantic search (Mukala et al., 2015) to include Natural Language processing techniques (NLP) for message curation and event logs construction. Just like in our previous work, we focus on the Initiation phase. During this phase, participants in Free/Libre and Open Source Software (FLOSS) engage with projects by reviewing and communicating, aiming to comprehend contents without making tangible contributions. The initiation stage is pivotal as participants access project repositories, exchange emails, and post messages to seek information or make requests. In Figure 1, we illustrate the transition of participants' activities from basic utilization to posting and making substantial contributions through commits during the practicing and developing phases.

In the initial paper, we presented an approach for mining the learning phases from these FLOSS development data sourced from the Openstack environment (OpenStack 2024). We consider the same data source for purpose of consistency in depicting differences, or/and improvements of our approach. Typical FLOSS repositories include archives from to CVS,bug reports, mailing archives and internet relay chats which all contain traces of learning activities. We previously stated that these FLOSS environments include specifically discussion forums and or emailing lists to which interested participants subscribe in order to follow a topic progression or to track progress in a feature they are interested in. Much of these discussions result in no substantial development work but are a part of the discussion for learning as they involve both Novice-Participants and Expert-Participants. Owing to their opensource and unregulated nature, these environments create a democratized space for learning and contributing.

Mukala (2015) extensively documented the literature pertaining to these environments, the different approaches and contributions of the landscape for knowledge generation and learning occurrences. A particular contribution has been the use of Process Mining as a way to explore, analyze and visualize the learning patterns from these environments as Learning Processes. Hence, our previous work (Mukala et al., 2014; Mukala et al., 2016; Mukala et al., 2017; Mukala

2025) has been largely focused on studying the learning activities derived from the FLOSS repositories using process mining techniques. Specifically, our previous work contributed in the learning of learning activities by visualizing and tracing the connections and flow of the learning activities as documented in the various FLOSS repositories or environments.

In this paper we present an approach considering the use of Machine Learning to learn and then infer the type of activity derived from Mailing Archives messages as they pertain to the Initiation Phase of the learning process. The type of machine learning used is a pretrained neural network called BERT which has been retrained on phased and activity tagged data from the Openstack FLOSS environment. Our distinct contribution in this paper is the use of Natural language processing to tag and to learn, predict with accuracy the learning activities of the Initiation Phase as they can be detected from exchanged messages, create the process models of learning activities.

The subsequent sections of the paper are organized as follows: Section 2 provides initial information on learning processes considerations (preliminaries) and log construction, offering a concise depiction of the Initiation Phase in the learning process. Section 3 delves into data collection and analysis, followed by the presentation of empirical results. Finally, Section 4 serves as the conclusion of the paper.

2. Preliminaries on Identifying Learning Activities in Mailing Archives

In order to identify activities and construct the event logs (required input for process mining) needed for our analysis, we undertake a number of tasks. The first task is analyzing the contents of emails. Text mining appears to be the most direct solution for this task as we need to analyze the contents of a post/email and deduct a corresponding activity. While we made use of Semantic Search in our initial endeavor (Mukala et al., 2025), the direction of this paper is on adopting NLP. Tracing learning activities requires semantic interpretation of email contents and this cannot be achieved by using any of these classical text mining tools, of which most are rule-based.

As outlined in our previous research findings and papers, this methodology relies on the utilization of specific key phrases for tagging and drawing inferences. Our selection of key phrases is grounded in various studies conducted within the Free/Libre and Open Source Software (FLOSS) domain, focusing on the types of questions and answers prevalent in FLOSS communication environments. Starting from this categorization, encompassing both question and response categories, we derive a set of key phrases. We try to incorporate all identified key phrases and expressions in the context of identifying learning activities and establishing the learning process across its three phases. It's important to note that in this paper, we specifically present the details of the first phase.

The creation of the initial event logs primarily relies on the utilization of a formal model depicting learning activities within Free/Libre and Open Source Software (FLOSS) communities. Additionally, lexical semantics plays a crucial role in compiling a list of key phrases relevant to our objectives. Lexical semantics involves exploring synonyms, homonyms, and the contextual meaning of words. Therefore, incorporating semantic search becomes essential, ensuring a comprehensive understanding of message contents in the identification of activities. Figure 2 illustrates a catalog that categorizes key phrases semantically, facilitating the identification of activities based on participants' roles in the Initiation Phase of the learning process.

The primary activities in this phase revolve around observation and establishing contacts to initiate the learning process. Ideally, this stage provides an opportunity for Novices to pose questions and seek assistance based on specific requests, with Experts intervening to respond to such inquiries. On one hand, Novices seeking help can engage in various activities, including Formulating Questions, Identifying Experts, Posting Questions, Commenting Posts or Messages, Contacting Experts, and Sending Detailed Requests. On the other hand, the main activities undertaken by Experts during the same period include Reading Messages on mailing lists or chat messages, Reviewing Posts from forums, Examining Source Code as participants commit code to the project, as well as Commenting Posts, Contacting Novices, and Commenting Posts.

STATES	GLOBAL KEYWORDS	PARTICIPANTS	ACTIVITIES	KEYPHRASE/CONDITIONAL ACTIVITY
Observation	"problem", "help", "error"	NOVICE	FormulateQuestion	If PostQuestion = true
			IdentifyExpert	"How did you do this", "I saw your code", "I need your help", "this does not work for me", "-1", "is this possible to do this", "can this be done?", "very helpful", "very well"
			PostMessage	If IdentifyExpert = true
			PostQuestion	"How can I do…?" "How to?", "don't understand how", "could help?", " what is wrong?", "my code is not running", "code not executing", "question", "How to", "what is wrong", "where can I", "Any ideas how to solve this problem?", "I have tried doing", " search for this", "but have had little luck", "any help?", "any suggestions?", "everything I could", "new to the"
			CommentPost	"does not work", "not executing", "this does not work for me", "do not know what is wrong with my code", "here is my code", "in short my problem", "step by step", "details provided", "works as follows", "I want it to", "expect it to", "my question is like this", "what I mean"
		EXPERT	ReadMessages	If CommentPost = true
			ReadPost	If CommentPost = true
			ReadSourceCode	"syntax error", "maybe you should…", "it seems to work for me""do not know what is wrong with the code" or "not sure it can"") or "running your code"
			CommentPost	"system details needed", "more details needed", "more problems details needed", "more details needed of what is on the screen", "Did it work before?", "provide exact step by step details"
ContactEstablishement	"can I get your help", "can you help", "send question", "contact details", "send email", "send file", "more details"	NOVICE	ContactExpert	If SendDetailedRequest = true
			SendDetailedRequest	"actually the code is like this…", "I tried this", "I don't know how", "I don't understand how?" "can you help", "your help", "you explain", "as you asked", "so my question is", "I wanted to know", "what I meant is", "my screenshot looks", "I get this error", "how do I fix this"
		EXPERT	ContactNovice	If CommentPost/SendFeedback = true
			CommentPost/SendFeedback	"does not work", "not executing", "maybe you should…", "it seems to work for me", "this does not look right" "you need to delete this..", "the syntax is not correct", "send me your code", "what is your problem?", "this works for me", "Did it work before", "I think it should work"

Figure 2. Catalog of key phrases for initiation phase

3. Data Collection and Analysis

Our data source is Openstack (OpenStack 2024). Just like with our initial work (Mukala et al. 2015), we considered the same environment for consistency and its fit-for-purpose aspect as we would like to look at differences that the previous approach and the one utilized in this paper have. OpenStack is a cloud computing platform that is both free and open-source. It is commonly utilized by users as an Infrastructure as a Service (IaaS) solution. This technology encompasses a collection of interconnected projects designed to oversee pools of processing, storage, and networking resources within a data center. Users have the ability to administer these resources via a web-based dashboard, command-line tools, or a RESTful API. OpenStack is released under the terms of the Apache License.

The resulting dataset procured from the Mailing List of Openstack contains a total of 36,204 emails from Nov 2018 to Dec 2023. The emails comprise of headers for each message sent. These headers are, from, to, date, subject, content, and Multipart.

The Content is our desired part of the dataset along with its date and time header as it allows us to create a timeline for our events log.

Our initial dataset cleaning process involved utilizing the BeautifulSoup library in Python for structural data manipulation. Given that the dataset messages contained a significant amount of HTML code and tags, we opted for this method to extract text from the HTML. Following this, we systematically removed all non-text symbols and remaining tags using regex formatting for identification and elimination.

Subsequently, we established a tabular data structure to manage and manipulate our dataset. The chosen structure for this purpose was the Pandas DataFrame, known for its table-like format and flexibility in manipulation. We then compiled a list of activities based on the catalog outlined in Figure 3, which had been previously employed in our earlier works. This catalog of keywords served as the guideline for activity identification, specifying the activity types for each participating resource, namely Expert and Novice.

To achieve this, we employed a set of activity keywords corresponding to each non-derived activity set, encompassing phrases relevant to each activity. These keywords play a pivotal role in assisting us in determining and deriving the activity type within the dataset. The identified activities were further categorized based on the participant types. These subsets of activities are subsequently utilized in Algorithm 3 of our work, wherein we delineate the steps necessary to assign an activity tag to each of these messages.

The primary innovation in this research lies in the application of Natural Language Processing (NLP) techniques to extract activities from a learned corpus, diverging from the Semantic Search method employed in the previous paper (Mukala et al. 2015). In pursuit of this goal, we opted for the Top2Vec algorithm, leveraging NLP models like BERT (Tenney et al. 2019), Doc2Vec from Gensim (Haider et al. 2020), and the Universal Sentence Encoder for NLP tasks. Specifically, our model of choice for this project was the Universal Sentence Encoder, selected for its accessibility of resources.

We retrained the model using our preprocessed dataset to discern embedded topics and generate embeddings and word vectors. These document embeddings were then employed to construct density maps, ultimately facilitating the identification of the closest words that closely match a given document. The selection of the Top2Vec algorithm was informed by several pertinent considerations. Notably, the algorithm operates without necessitating a predefined list of stop words and lemmatizations, tasks that are fundamental yet intricate in NLP studies. Additionally, it effectively generates topics from a corpus and furnishes search functions that prove highly valuable. These factors collectively influenced our decision to adopt this algorithm. While we acknowledge the existence of potentially superior algorithms that could yield better results, we emphasize that their application in generating event logs has not been demonstrated in prior works (Table 1).

Table 1. Algorithm 1: Main Construct Activity Dataframe

Algorithm 1: Main Construct Activity Dataframe

Input**: **Activity_Name**, **dataFrame_Result_Collector
***Output:** dataFrame with Event Log*
 # Create Activity list
***ActivityList** ← listOfActivityNames*

 # Create Searching Keywords Lists
***ActivityTypeKeywordList** ← List Of Keywords from given table*

 # Create Activity Dictionary
***AcitivityDict** ← {**Activity**: **Activity_keywords**}*

***ParticipantActivityDictionary** = { **participant**: **ActivityList** }*
 # Ex. activity_participant = {'novice':novice_activity}

1 *# Create Model*
***Model** ←**Top2Vec**(content,embedding_model,vocab.connector_words)*

2 *# Find the documents and ID for Activity*
For** Phrase **IN** Activity_Dict[Activity_name] **do

3 ***Documents**,*
***document_score**,*
***document_ID** ← **Model**.SEARCH_BY_KEYWORDS(Phrase)*

4 ***end***
5 ***For** Each Document, Document_Score, Document_ID **do***
6 ***dataFrame_INTERNAL** ← **create_DataFrame**()*
***dataFrame** ← **set_dataFrame_activity**(dataFrame, Document_ID, Activity_name)*
***dataFrame_INTERNAL** ← **create_activity** (dataFrame, Document_ID)*
***dataFrame_RESULT_COLLECTOR**.append (dataFrame_ _INTERNAL)*

7 ***end***

```
1
2 model = Top2Vec(df.cleaned_content.values,
3                 embedding_model='universal-sentence-encoder',
4                 ngram_vocab=True,
5                 ngram_vocab_args={"connector_words": "phrases.ENGLISH_CONNECTOR_WORDS"},)
6
7
```

Figure 3. Python Code Snippet for model creation

Table 2. Algorithm 2: Construct Participant Column

Algorithm 2: Construct Participant Column

Input: *Participant, ParticipantActivityDictionary, dataFrame_RESULT_COLLECTOR, Activity_Name*
Output: *dataFrame with Event Log And Participant Column*

```
For Row, Index IN dataFrame_RESULT_COLLECTOR do
    For Activity IN ParticipantActivityDictionary[Participant] do
        IF dataFrame_RESULT_COLLECTOR.[Index, Activity] == Activity_Name do
            dataFrame_RESULT_COLLECTOR[Index, ParticipantColumn] =
            Participant
        end
    end
```

Subsequently (Table 2 and Table 3), we conducted a search for every phrase linked to each activity. To illustrate, for the initial activity "IdentifyExpert," we employed all the phrases associated with this particular activity to query our embedded documents. The resulting documents and their corresponding indexes were then utilized to tag and correlate the specified activity with the respective emails. For organizational purposes, the activities and their tags were stored in distinct Dataframes for each discovered document. These individual Dataframes were later compiled and saved separately, capturing the indexes and activities for all non-derived activities listed in our activity catalog (Figure 4)

```python
def create_activity(df,idx):
  df_appended = pd.DataFrame()
  df_new = pd.DataFrame()
  df_new.at[idx,'Date'] = df.iloc[idx]['Date']
  df_new.at[idx,'Message-ID'] = df.iloc[idx]['Message-ID']
  df_new.at[idx,'Case-ID'] = df.iloc[idx]['Message-ID'] + str(idx)


  if df.iloc[idx]['PostQuestion'] == 'True':
    df.at[idx,'FormulateQuestion'] = 'True'

    df_new = set_df_activity_col(df_new,idx, 'PostQuestion')
    df_appended = data_frame_append(df_appended, df_new)

    df_new = set_df_activity_col(df_new,idx,'FormulateQuestion')
    df_appended = data_frame_append(df_appended, df_new)

```

Figure 4. Python Code Activity Generation

Following these procedures, we achieved the capability to generate event logs encompassing the entire mailing archive. Our definition of an event log comprises a sextuple arranged in the following order: (serial number, Date, Message-ID, Case-ID, Activity, Participant).

Table 3. Algorithm 3: Derive Activity based on Rules

Algorithm 3: Derive Activity based on Rules

***Input**: dataFrame, Index*
***Output**: DataFrame with Activity Column Set to Activities Derived*

```
# Create two new dataframes
dataFrame_NEW, dataFrame_APPENDED ← CreateNewDataFrame()
dataFrame_NEW ← dataFrame[Date, Message-ID]
dataFrame_NEW[Case-ID] ← dataFrame[Message-ID] + Index

For Each Activity With Keyword do

    # Check if this Activity Column is Set at Index

    IF dataFrame(Index, Activity) == True do
        # Set Derived Activty Column in dataFrame
        dataFrame[ Index , Derived_activity ] = True

        # At this index set the activity name as below
        dataFrame_NEW ← set_dataFrame_activity(dataFrame_NEW, index,
        Activity)
        dataFrame_appened ← copy_dataFrame(dataFrame_new)


        # At this index set derived activity
        dataFrame_NEW ← set_dataFrame_activity(dataFrame_NEW, index,
        DERIVED_Activity)

        dataFrame_appened ← copy_dataFrame(dataFrame_new)

    end
end
```

The resulting event log adheres to the required format for analysis and visualization using Disco (Günther and Rozinat 2012). Leveraging Disco, we crafted Process Models that illustrate the occurrence of learning activities as documented by their corresponding email messages.

Disco facilitates the representation of models through a graphical workflow and provides enriched statistical information for in-depth analysis. It allows for a process model to be portrayed with frequency metrics elucidating the flow of event occurrence. The primary aim of these frequency metrics is to showcase how often specific parts of the processes have been executed, encompassing absolute frequency, case frequency, and maximum repetitions.

These metrics are employed to model learning activities executed by both Novices and Experts. While additional numerical measures like events over time, active cases during the specified period, case variants, the number of events per case, and case duration can be plotted as necessary, for the sake of simplicity and efficiency, we choose to represent only key statistical details. These details are deemed most representative of the presence, impact, and occurrence of learning activities in FLOSS over the selected period.

4. Results and Discussion

4.1 Process Models from NLP

In presenting our findings, it's essential to outline the structural aspects of our results. The email archives utilized in our investigation span from November 18, 2018, to December 27, 2023. These emails originate from the OpenStack Discuss mailing archives, serving as a platform dedicated to both discussion and the ongoing development of the OpenStack software, providing valuable insights into its recent evolution.

Figures 5 through 7 visually illustrate as Process Models the events documented in these emails, with each event represented by blocks. The coloration of these blocks signifies the relative frequency of the corresponding events. In total, our experimental efforts generated 49,423 events. The cases, totaling 4,252, each encompass multiple events.

Breaking down the total events, it's noteworthy that Novice activities slightly surpass Expert activities, constituting 52.3% or 25,834 events of the overall total.

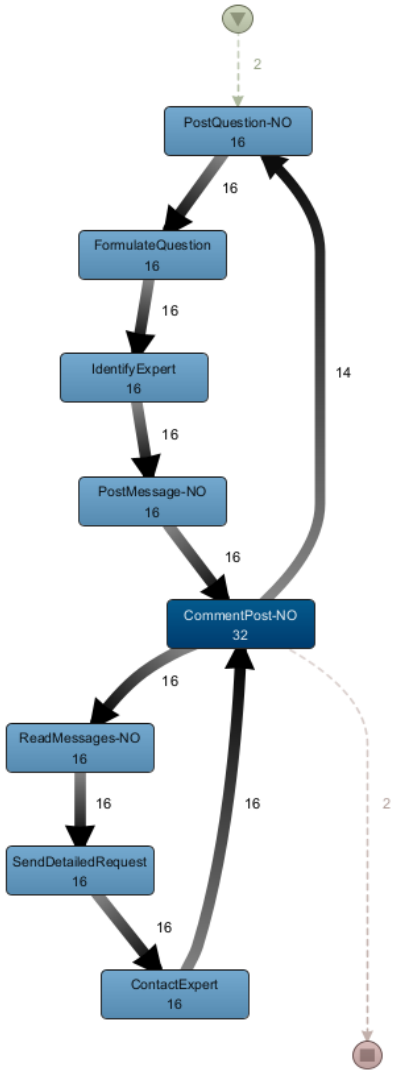

Figure 5. Process Model for Novice–Per frequency [Initiation Phase] in Mailing Lists of OpenStack

These process models illustrate a sequential arrangement or list of activities initiated by a resource, typically an individual involved in a case. In this context, a case serves as an index for an event log, with resources contributing

activities to it. The process models, as depicted in Figure 5 and Figure 6, represent the Novice and Expert participants, respectively.

A discernible pattern of steps associated with each resource becomes evident from these logs. The Novice participant initiates the process by creating and formulating a question. A key subsequent event for Novices involves posing a question aimed at identifying an expert in the relevant topic. Following the posting of the message on the mailing list, Novices either engage in reading the messages within the thread or receive feedback from the Expert, as depicted in Figure 6.

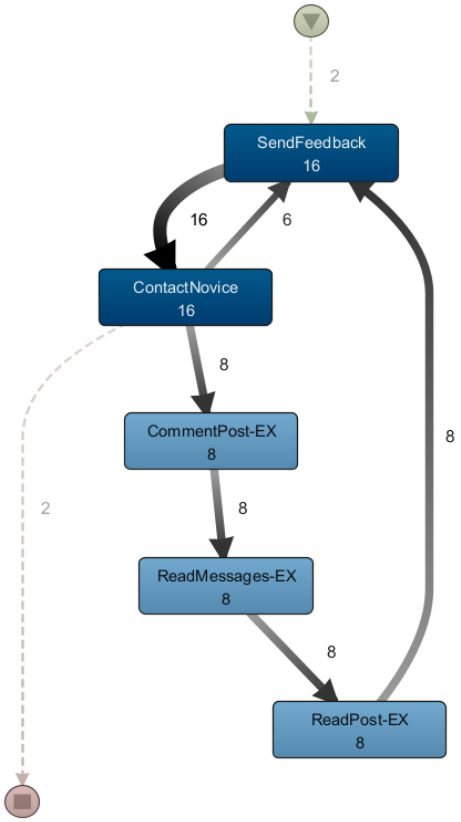

Figure 6. Process Model for Expert–Per frequency [Initiation Phase] in Mailing Lists of OpenStack

Our generated event logs show a typical sequential pattern for the Expert as well. As depicted in Figure 6, the Expert initiates the process by Sending Feedback on the Post. Subsequently, they either establish direct contact with the Novice or provide detailed Comments on the post. It's important to note that the act of sending feedback doesn't necessarily imply the posting of a comment; rather, feedback can take the form of instructional phrases commonly used. In contrast, a comment engages in comprehensive discussion and communication.

These events occur repeatedly, forming a loop of discussion and learning, which is of particular interest to us. In an empirical manner, these events illustrate the sequence of activities within a learning process in the context of software development. It's worth noting that an Expert can, at times, also function as a seeker of knowledge, such as when seeking information from other Experts in a topic outside their specialty, thereby assuming the role of a Novice.

4.2 Process Models from NLP vs Process Models from Semantic Search (Mukala et al. 2015)

Examining Figure 7 and Figure 8 below, the positive outcome is that employing both approaches on the same dataset enabled us to discern learning activities from participants during the Learning phase under consideration. The notable distinction lies in both the frequency of activity occurrences and the presence of distinct clusters showcasing patterns of activities.

Primarily, a significant difference is observed compared to our previous work, where our model illustrated patterns with two initiating branches: one stemming from "FormulateQuestion" and another from "CommentPost." Although the subsequent activities may not vary significantly, the key observation is the existence of potential clusters of users in the dataset exhibiting two distinct sets of behaviors. It can be inferred that, through Semantic search, the visualization of learning patterns is neither deterministic nor static. Instead, these patterns occur in various sequences, as demonstrated by the different activities we identify.

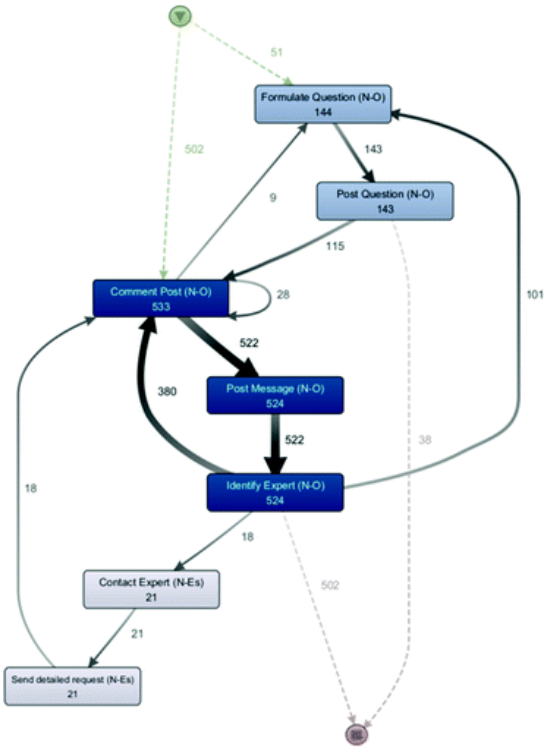

Figure 7. Process Model for Novice–Per frequency [Initiation Phase] in Mailing Lists of OpenStack using Semantic Search (Mukala et al. 2015)

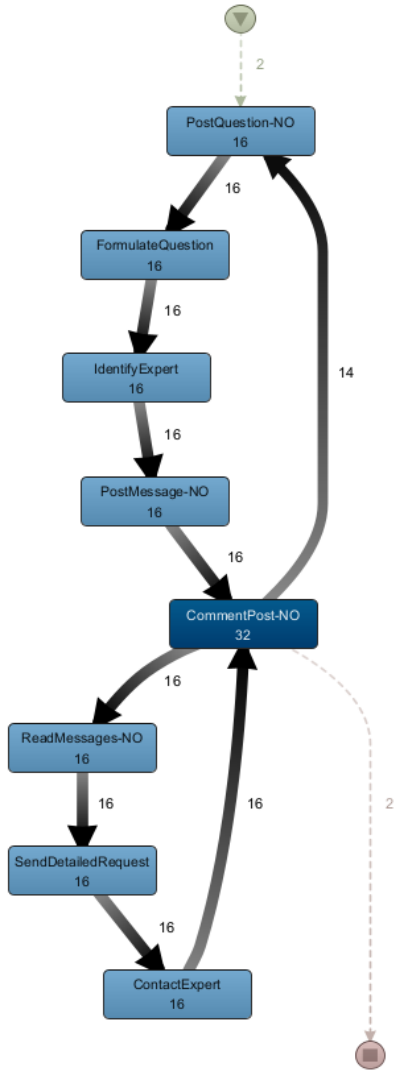

Figure 8. Process Model for Novice–Per frequency [Initiation Phase] in Mailing Lists of OpenStack using NLP

In contrast, utilizing Natural Language Processing (NLP) yields a significantly enhanced and coherent pattern of learning that seems to encapsulate the entire dataset. A singular starting point emerges, portraying behaviors indicative of a generalized sense of learning for participants engaged in these learning types. Building upon our prior work (Mukala 2015), we emphasized the importance of considering the expected maturation level in a learning process. Certain activities prevail in the initial stages, reflecting the level of competency, and as knowledge accumulates, participants demonstrate a steady progression in their learning journey.

The use of NLP provides an additional advantage in terms of accuracy and near-precise detection of this trend, allowing for graphical representation to validate our initial observations.

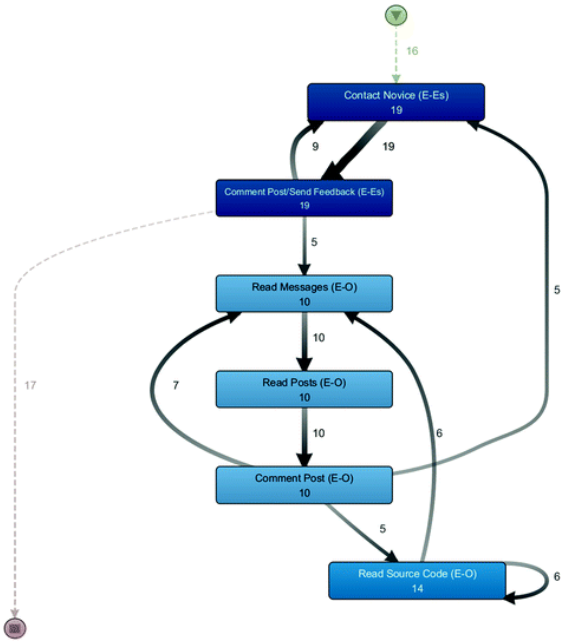

Figure 9. Process Model for Expert–Per frequency [Initiation Phase] in Mailing Lists of OpenStack using Semantic Search (Mukala et al. 2015)

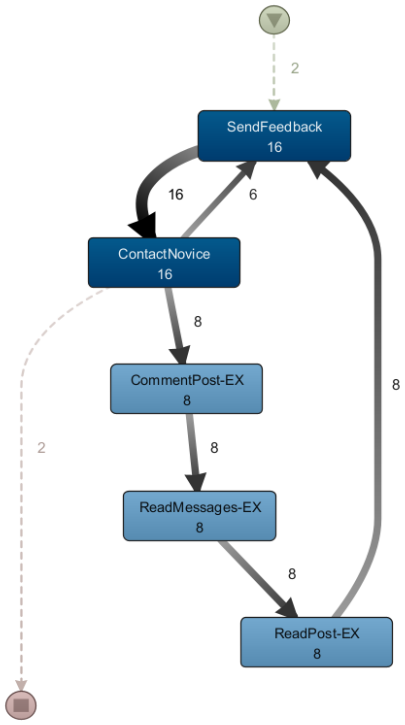

Figure 10. Process Model for Expert–Per frequency [Initiation Phase] in Mailing Lists of OpenStack using NLP)

A parallel observation arises when considering the learning behavior of the Expert, identified by both Semantic Search and NLP, as depicted in Figures 9 and 10. Accurately identifying these activities at this stage of the learning process is easily achievable using both approaches. While the sequence of activities is largely maintained, it becomes apparent that NLP provides a more robust and generic categorization of learning events compared to Semantic Search, which tends to encompass multiple clusters, as observed with Novices.

It is undeniable that both approaches offer possibilities for text mining to derive learning activities, but NLP seems to present a more accurate and comprehensive representation of learning processes, as depicted in the theoretical formalizations.

5. Conclusion

This paper, along with our previous ones, collectively emphasizes that Free Libre Open Source Software (FLOSS) environments appear to offer valuable learning opportunities for participants. While literature supports this notion,

there has been limited empirical work in this domain. Existing studies have predominantly relied on content data gathered through surveys, questionnaires, or reports from community observers with a defined period of involvement. Research indicates a growing number of participants actively engaging in these platforms through discussions and email exchanges, generating substantial volumes of data that inherently contain evidence of learning.

Since learning activities are not directly observable in the repositories, we expanded upon our prior work on semantic search by integrating the advantages offered by Artificial Intelligence (AI) through Natural Language Processing (NLP). This integration aims not only to confirm the existence of such activities (learning patterns) in these environments but also to empirically substantiate their presence through text mining.

By employing a blend of Natural Language Processing models, key phrases, and a set of rules, our approach has not only reiterated but expanded upon our prior findings, shedding further light on how to uncover traces of interaction and learning activities in Free Libre Open Source Software (FLOSS) email messages. The feasibility of tracing these activities has been established, with process mining emerging as a catalytic tool for identifying such activities within FLOSS. Our work contributes by offering empirical evidence regarding the existence of learning processes in FLOSS environments, as demonstrated through the analysis of OpenStack Mailing Archives.

This paper underscores that, through the integration of Artificial Intelligence (AI) and Natural Language Processing (NLP), we can attain results that are notably more accurate and closely aligned with the formalism of learning patterns found in the existing literature. Our intention is to extend this research by applying the same approach to the next two phases of the learning process and potentially expanding it to other FLOSS environments. This expansion aims to uncover main patterns while also identifying potential differences, thereby enhancing our understanding of these patterns and providing a robust foundation for knowledge generation. Such insights can significantly contribute to shaping policies surrounding the adoption of participation in FLOSS projects, offering a viable alternative for imparting practical skills to computer science and programming students.

References

Bacon S, Dillon T. The potential of open source approaches for education. Futurelab. 2006;44.

Cerone A, Sowe SK. Using free/libre open source software projects as e-learning tools. Electronic Communications of the EASST. Dec 10;33, 2010.

Cerone A. Learning and activity patterns in OSS communities and their impact on software quality. Electronic Communications of the EASST 48, 2013.

Fernandes S, Cerone A, Barbosa LS. A preliminary analysis of learning awareness in FLOSS projects. InInformation Technology and Open Source: Applications for Education, Innovation, and Sustainability: SEFM 2012 Satellite Events, InSuEdu, MoKMaSD, and OpenCert Thessaloniki, Greece, October 1–2, 2012 Revised Selected Papers 10 2014 (pp. 133-139). Springer Berlin Heidelberg, 2014.

Fernandes, S., Cerone, A., & Barbosa, L. S. FLOSS Communities as Learning Networks. Int. Journal of Information and Education Technology, 3(2), pages 278–281, IACSIT Press, April 2013

Günther CW, Rozinat A. Disco: Discover Your Processes. BPM (Demos). 940(1):40-4, 2012.

Haider MM, Hossin MA, Mahi HR, Arif H. Automatic text summarization using gensim word2vec and k-means clustering algorithm. In2020 IEEE Region 10 Symposium (TENSYMP), (pp. 283-286). IEEE, 2020.

Jaccheri L, Osterlie T. Open source software: A source of possibilities for software engineering education and empirical software engineering. InFirst International Workshop on Emerging Trends in FLOSS Research and Development (FLOSS'07: ICSE Workshops 2007) (pp. 5-5). IEEE, 2007.

Meiszner A, Glott R, Sowe SK. Free/Libre Open Source Software (FLOSS) communities as an example of successful open participatory learning ecosystems. UPGRADE, The European Journal for the Informatics Professional. Jun; 9(3):62-8, 2008.

Meiszner, A., Mostaka, K., & Syamelos, I. A hybrid approach to Computer Science Education–A case study: Software Engineering at Aristotle University, Greece, 2009.

Mukala P, Buijs JC, Leemans M, van der Aalst W. Learning analytics on coursera event data: A process mining approach. In5th International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA 2015 (pp. 18-32), 2015. CEUR-WS. Org.

Mukala P, Buijs JC, Van Der Aalst WM. Exploring students' learning behaviour in moocs using process mining techniques, Technical Report, Eindhiven University of Technology, The Netherlands, 2015.

Mukala P, Cerone A, Turini F. An abstract state machine (ASM) representation of learning process in FLOSS communities. InSoftware Engineering and Formal Methods: SEFM 2014 Collocated Workshops: HOFM, SAFOME, OpenCert, MoKMaSD, WS-FMDS, Grenoble, France, September 1-2, 2014, Revised Selected Papers 12 2015 (pp. 227-242). Springer International Publishing, 2015.

Mukala P, Cerone A, Turini F. An empirical verification of a-priori learning models on mailing archives in the context of online learning activities of participants in free\libre open source software (FLOSS) communities. Education and Information Technologies. 22:3207-29, 2017.

Mukala P, Cerone A, Turini F. Mining learning processes from FLOSS mailing archives. InOpen and Big Data Management and Innovation: 14th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society, I3E 2015, Delft, The Netherlands, October 13-15, 2015, Proceedings 14 (pp. 287-298), 2015. Springer International Publishing.

Mukala P, Cerone A, Turini F. Ontolifloss: Ontology for learning processes in FLOSS communities. InSoftware Engineering and Formal Methods: SEFM 2014 Collocated Workshops: HOFM, SAFOME, OpenCert, MoKMaSD, WS-FMDS, Grenoble, France, September 1-2, 2014, Revised Selected Papers 12 (pp. 164-181). Springer International Publishing, 2015.

Mukala P, Cerone A, Turini F. Process mining event logs from FLOSS data: State of the art and perspectives. InSoftware Engineering and Formal Methods: SEFM 2014 Collocated Workshops: HOFM, SAFOME, OpenCert, MoKMaSD, WS-FMDS, Grenoble, France, September 1-2, 2014, Revised Selected Papers 12 2015 (pp. 182-198). Springer International Publishing, 2015.

Mukala P. A Temporal Visual Distribution of Learning Activities in FLOSS Repositories. Technical Report, University of Pisa, Italy, 2016.

Mukala P. Mining Reviews in Open Source Code for Developers Trail: A Process Mining Approach. arXiv preprint arXiv:2308.00686. Jul 16, 2023.

Mukala P.. Process Models for Learning Patterns in FLOSS Repositories. PhD thesis, Department of Computer Science, University of Pisa, 2015.

OpenStack. In Wikipedia, The Free Encyclopedia. Retrieved 15:48, January 10, 2024, from http://en.wikipedia.org/w/index.php?title=OpenStack&oldid=632224644 S. (2024, January 3)

Papadopoulos PM, Stamelos IG, Meiszner A. Enhancing software engineering education through open source projects: Four years of students' perspectives. Education and Information Technologies. 18:381-97, 2013.

Sowe SK, Stamelos I, Deligiannis I. A framework for teaching software testing using F/OSS methodology. InOpen Source Systems: IFIP Working Group 2.13 Foundation on Open Source Software, June 8–10, 2006, Como, Italy 2 (pp. 261-266). Springer US, 2006.

Sowe SK, Stamelos I. Reflection on knowledge sharing in F/OSS projects. InOpen Source Development, Communities and Quality: IFIP 20 th World Computer Congress, Working Group 2.3 on Open Source Software, September 7-10, 2008, Milano, Italy 4 (pp. 351-358). Springer US, 2008.

Sowe SK, Stamelos IG. Involving software engineering students in open source software projects: Experiences from a pilot study. Journal of Information Systems Education. 2007 Nov 1;18(4):425, 2007.

Tenney I, Das D, Pavlick E. BERT rediscovers the classical NLP pipeline. arXiv preprint arXiv:1905.05950. May 15, 2019.

Biographies

Patrick Mukala is an Assistant Professor of Artificial Intelligence and Data Science. With a background and training in Computer Science, He teaches computer science courses and conduct research pertaining to AI and Data Engineering. He holds a double PhD in Computer Science. His research interests are in AI for Data and Process analytics (Applied Analytics) and Software Engineering. In Applied Analytics, his interests are in Learning Analytics, Process-Centric Analytics, Healthcare Analytics and Governance analytics. The objectives of these research endeavors are centered around understanding and determining the appropriateness of analytics techniques regarding current data analysis demands as demonstrated by issues in the real world; developing and providing novel process-centric paradigms for data analysis and their application; as well as working in close collaboration with industry to understand the demands and provide timely and relevant solutions thereto. He also does extensive work in semantic web, abstract state machines, model checking and automata theory. Dr Mukala has numerous years in research and teaching from the Tshwane University of Technology in South Africa, the Technical University of Eindhoven, and the Fontys University of Applied sciences in the Netherlands. For over 5 years, He worked for several companies in the Netherlands in various capacities in data architecting, data engineering, data science and AI.

Obaid Ullah is an instructor at UOWD Faculty of Computer Science. He teaches various comptuer Sciecne courses at UOWD including AI and Data analytics with Professor Mukala. He holds Masters degree in Computer Science from Ghulam Ishaq Khan Insttitue, Pakistan and Bachelor degree in Computer Systems Engineering. He was involved in Autonomous Vehicles Mobility , 5G, Computer Vision and Deeplearning at the Emirates Center For Mobility Research at United Arab Emirates University as a Research Assistant. He has worked professionally as a Machine Learning Engineer at Stockholm, Sweden. His research interests include Reinforcement Learning and Computer Vision.